

BEST AVAILABLE COPY



IN THE UNITED STATES
PATENT AND TRADEMARK OFFICE

APPLICANT: Kerry Champion
APPLICATION NO.: 10/015,502
FILING DATE: December 11, 2001
TITLE: Traffic Manager for Distributed Computing Environments
EXAMINER: Pramila Parthasarathy
GROUP ART UNIT: 2136
ATTY. DKT. NO.: 23982-11569

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Mail Stop Amendment, Commissioner For Patents, P. O. Box 1450, Alexandria, VA 22313-1450, on the date shown below:

Dated: June 23, 2006

By: Nubia Morad

Nubia Morad

MAIL STOP AMENDMENT
COMMISSIONER FOR PATENTS
P.O. BOX 1450
ALEXANDRIA, VA 22313-1450

DECLARATION OF FACT BY KERRY CHAMPION UNDER 37 C.F.R. § 1.131

I, Kerry Champion, hereby declare the following:

1. I am the sole inventor of the invention described and claimed of U.S. Patent Application Serial No. 10/015,502 (hereinafter "Subject Application"), entitled "Traffic Manager for Distributed Computing Environments," filed on December 11, 2001.

2. I conceived of and reduced to practice the invention – as described and claimed in claims 1 through 68 as filed in the Subject Application (hereinafter "the claimed invention") – in this country before September 21, 2001. My prior conception and reduction to practice are evidenced by the following:

a. Attached hereto as Exhibit A is a true and correct copy of selected pages of PowerPoint presentation showing an implementation of a SOAP traffic manager. As can be seen, the SOAP traffic manager enables the exchange of SOAP messages between client and server programs. In addition, the SOAP traffic manager enables security models (such as encryption/decryption and signature verification), as well as malicious attack protections (such as service attacks) and DMZ policy enforcement (DMZ stands for demilitarized zone, which is an area that exists between trusted and untrusted networks to provide additional levels of security). The SOAP traffic manager is shown: receiving a SOAP message, determining whether a security rule has been defined for the SOAP message (based on a security policy for exchanging SOAP messages), and performing a security related operation on the SOAP message based on the security rule. I gave this presentation on September 11, 2001, as further indicated by the date in the lower left-hand corner of the selected pages. I prepared this presentation well in advance of that date.

b. Attached hereto as Exhibit B is a true and correct copy of a "SOAP Traffic Manager Software Architecture Document." The purpose of this document was to enable one skilled in the art to implement an operational embodiment of the SOAP traffic manager. As can be seen, the described SOAP traffic manager enables the exchange of SOAP messages between client and server programs. The SOAP traffic manager, among other things, defines transformations (e.g., for mapping between a key or other security identifier used by a client program and a key or other security identifier used by a server program) and assigns rights for a particular SOAP interface (e.g., when a SOAP interface should be published or accessed by a particular individual, based on a rule associated with that SOAP interface). The Use Case, Logical, Deployment, and Data Views of the Architecture Document readily demonstrate to one skilled in the art how to implement a SOAP traffic manager capable of: receiving a SOAP message, determining whether a security rule has been defined for the SOAP message (based on a security policy for exchanging SOAP messages), and performing a security related operation (e.g., encryption/decryption, digital signature signing/verification, granting/denying interface publication/access, and assess whether the SOAP message constitutes a service attack) on the SOAP message based on the security rule. One specific example provided in the

Architecture Document is under section 6.1, on page 9, which describes the policy: "If the service Withdrawcredit is called with parameter Amount greater than 10000 then send mail to Administrator and fail request." Here, a received SOAP message is attempting to access the Withdrawcredit service (provided by some server) via a SOAP interface of the SOAP traffic manager. However, there is a rule associated with that SOAP message/interface which states: if the parameter Amount is greater than 10000, then certain operations must be performed. In particular, an email is sent to the Administrator, and the client request is failed (i.e., no access is granted to the user for that particular service/SOAP interface). The last noted revision date of the Architecture Document is September 3, 2001 (shown in European format as "03.09.2001"). The 1999 copyright date is not applicable.


c. Exhibits A and B illustrate my conception of a SOAP traffic manager as defined by the claimed invention, as discussed with points a and b above.

d. Computer implemented methodologies, modules, and apparatuses are commonly implemented based on Block diagrams, Use Case Views, Logical Views, Deployment Views, and Data Views, such as those shown in Exhibits A and B. Thus, these diagrams and views (as well as their accompanying written description), further demonstrate that the claimed invention was reduced to practice before September 21, 2001.

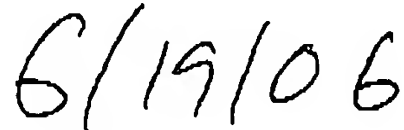
e. From at least September 21, 2001 through the preparation and filing of the Subject Application, I diligently worked to communicate the claimed invention to a patent attorney for the purpose of filing a patent application, resulting in the Subject Application prepared and filed with due diligence.

3. Therefore, the claimed invention was both conceived and reduced to practice before September 21, 2001.

4. I hereby declare that all statements made herein to the best of my own knowledge are true and that all statements made on information and belief are believed to be true; that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine and/or imprisonment under 18 U.S.C. § 1001; and that such willful statements may jeopardize the validity of the application or any patent issued thereon.

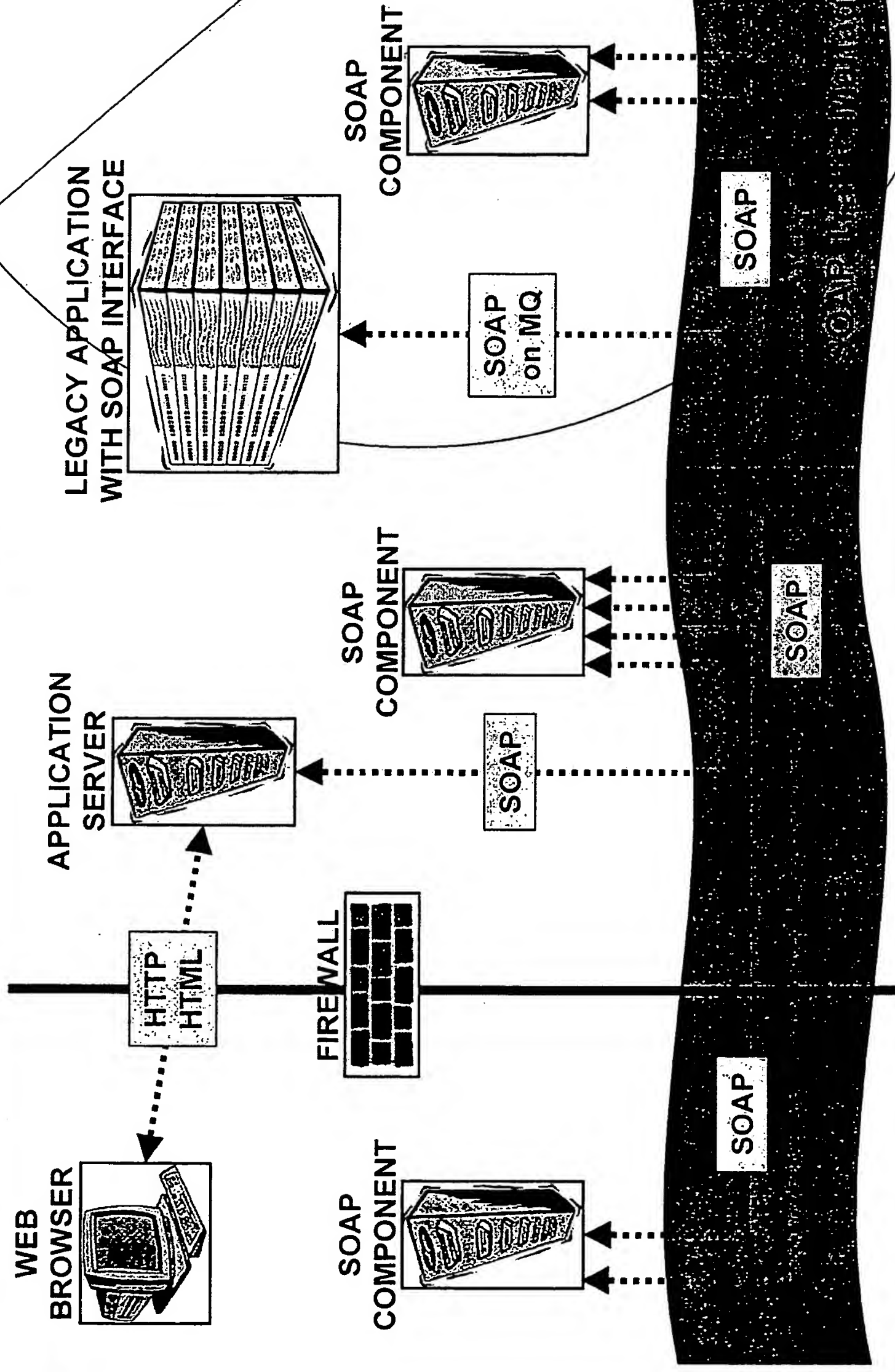


Kerry Champion



Date

Stele: SOAP Traffic Manager



Environmental Differences

- Security models
- User and role directories
- Schema definitions (data dictionaries)
- Underlying transport
 - HTTP, HTTPS, HTTPR, SMTP, MQ, EDI VANS, etc.
- Services interfaces directories (UDDI)
- Version upgrade cycles
- Performance optimization assumptions

Organizational Policies

- Audit trails and archiving
- Directory and data access limitations
- Move partner controls to the edges
- Malicious attack protections
- Legal and regulatory enforcement
- Data release, review and approval
- API availability, review and approval
- API and documentation convention ^{intermediate} compliance ^{IT Developer} *get it off his mind*
- DMZ policy enforcement

Stele

**Stele SOAP Traffic Manager
Software Architecture Document**

Version <1.0>

Stele SOAP Traffic Manager	Version: 1.0
Software Architecture Document	Date: 03.09.2001
<document identifier>	

Revision History

Date	Version	Description	Author
03.09.2001	1.0	Initial Version	Konstantin Vassilev

Stele SOAP Traffic Manager	Version: 1.0
Software Architecture Document	Date: 03.09.2001
<document identifier>	

Table of Contents

1. Introduction	4
2. Architectural Goals and Constraints	4
3. Use-Case View	4
4. Logical View	4
4.1 Main	5
4.2 Services	5
4.3 Users	6
4.4 Policies	7
5. Deployment View	7
6. Data View	8
6.1 Policy definition format	9
6.1.1 Conditions	9
6.1.2 Actions	10

Stele SOAP Traffic Manager	Version: 1.0
Software Architecture Document	Date: 03.09.2001
<document identifier>	

Software Architecture Document

1. Introduction

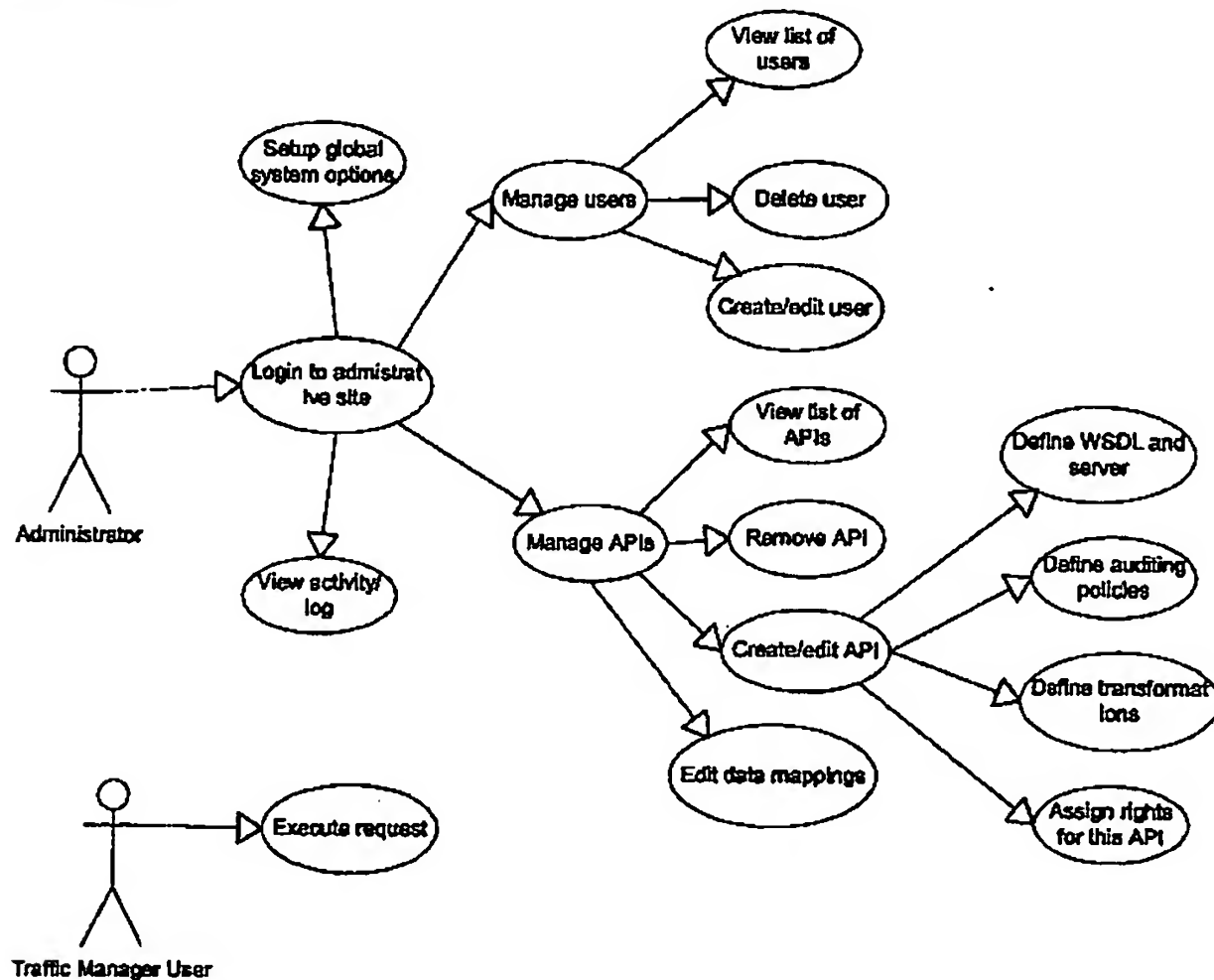
This document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.

2. Architectural Goals and Constraints

The architecture of the Stele SOAP Traffic Manager aims to achieve the following goals:

- Platform independence
- DB platform independence
- Scalability
- Maximum re-use of 3rd party supported components

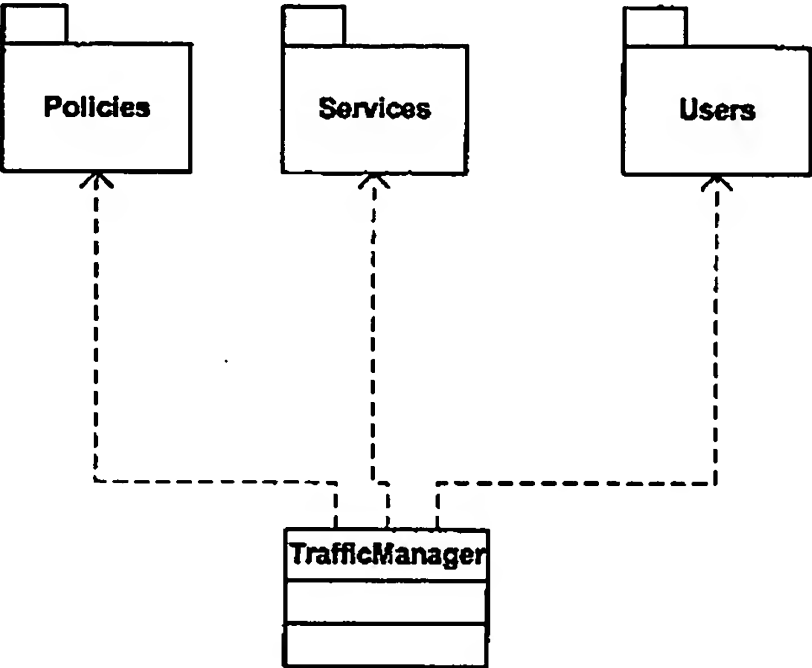
3. Use-Case View



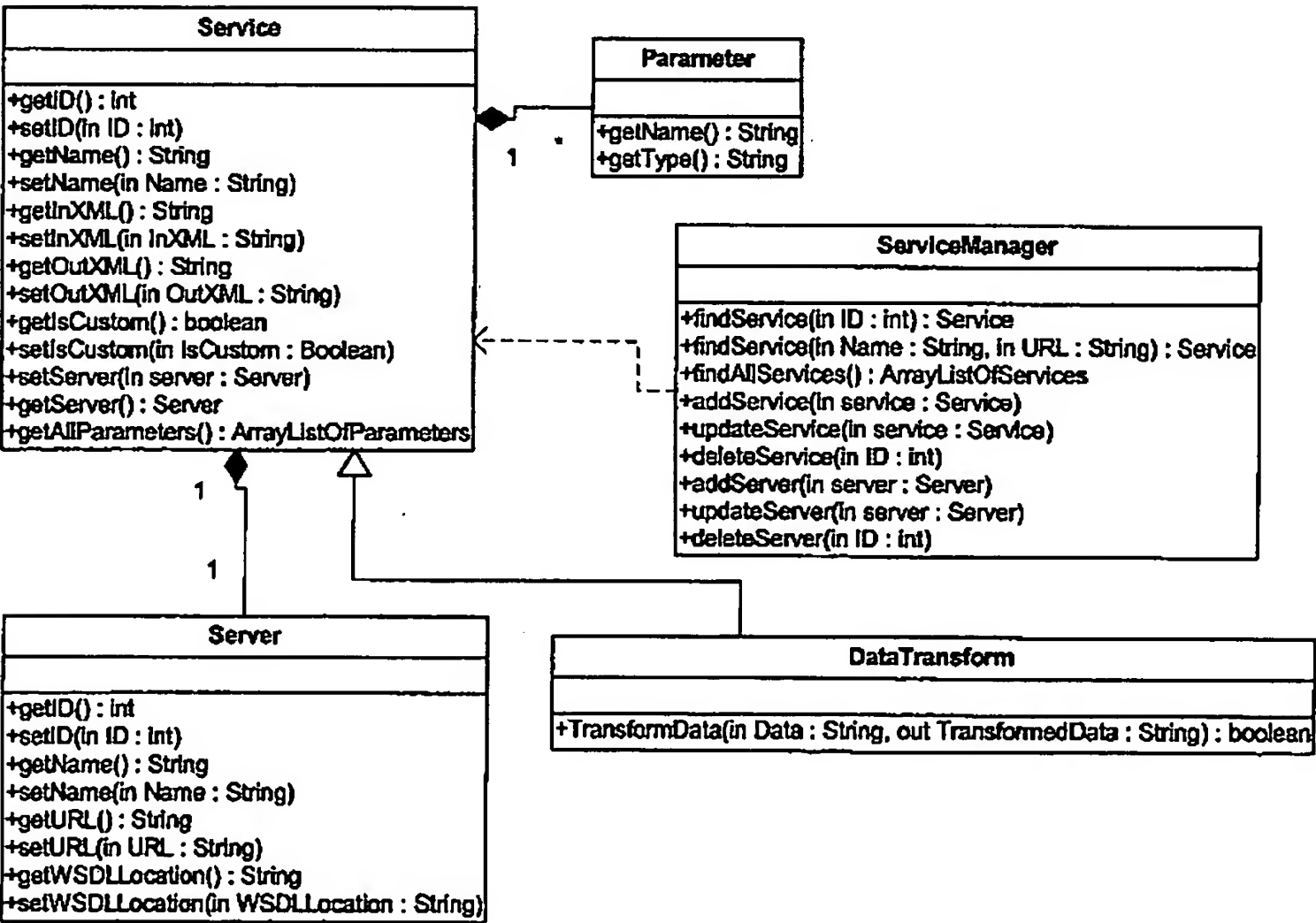
4. Logical View

Stele SOAP Traffic Manager	Version: 1.0
Software Architecture Document	Date: 03.09.2001
<document identifier>	

4.1 Main

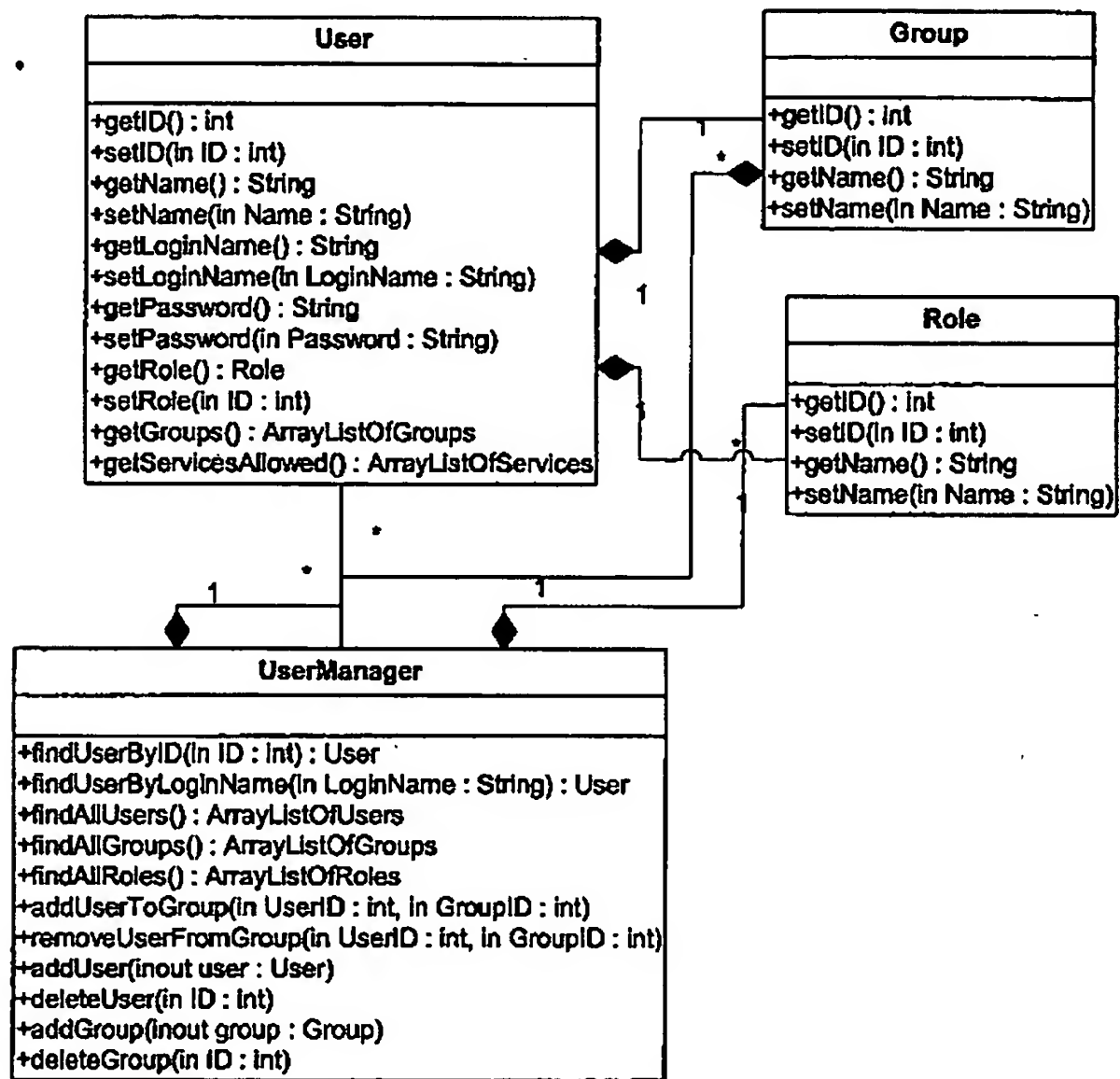


4.2 Services



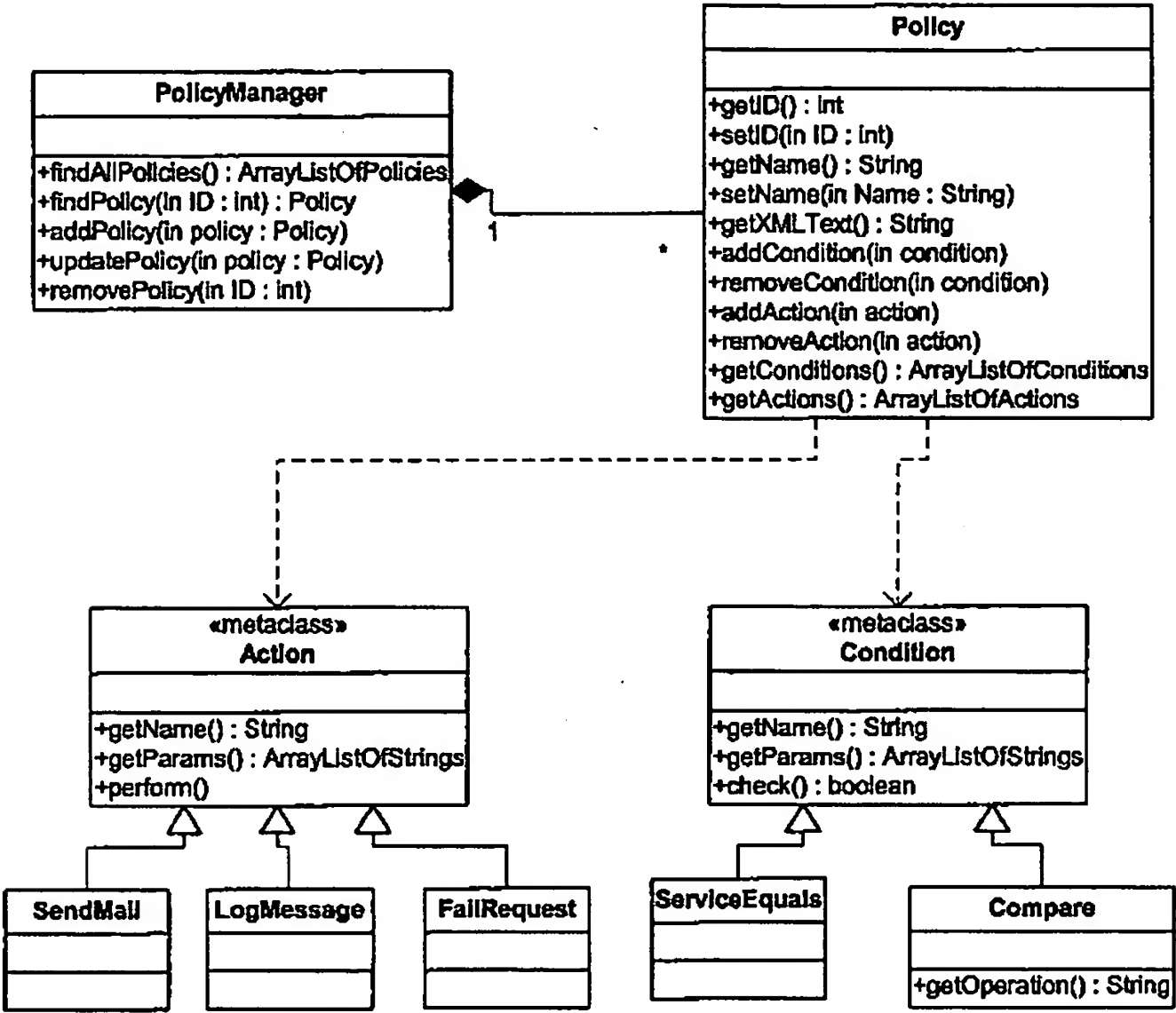
Stele SOAP Traffic Manager	Version: 1.0
Software Architecture Document	Date: 03.09.2001
<document identifier>	

4.3 Users



Stele SOAP Traffic Manager	Version: 1.0
Software Architecture Document	Date: 03.09.2001
<document identifier>	

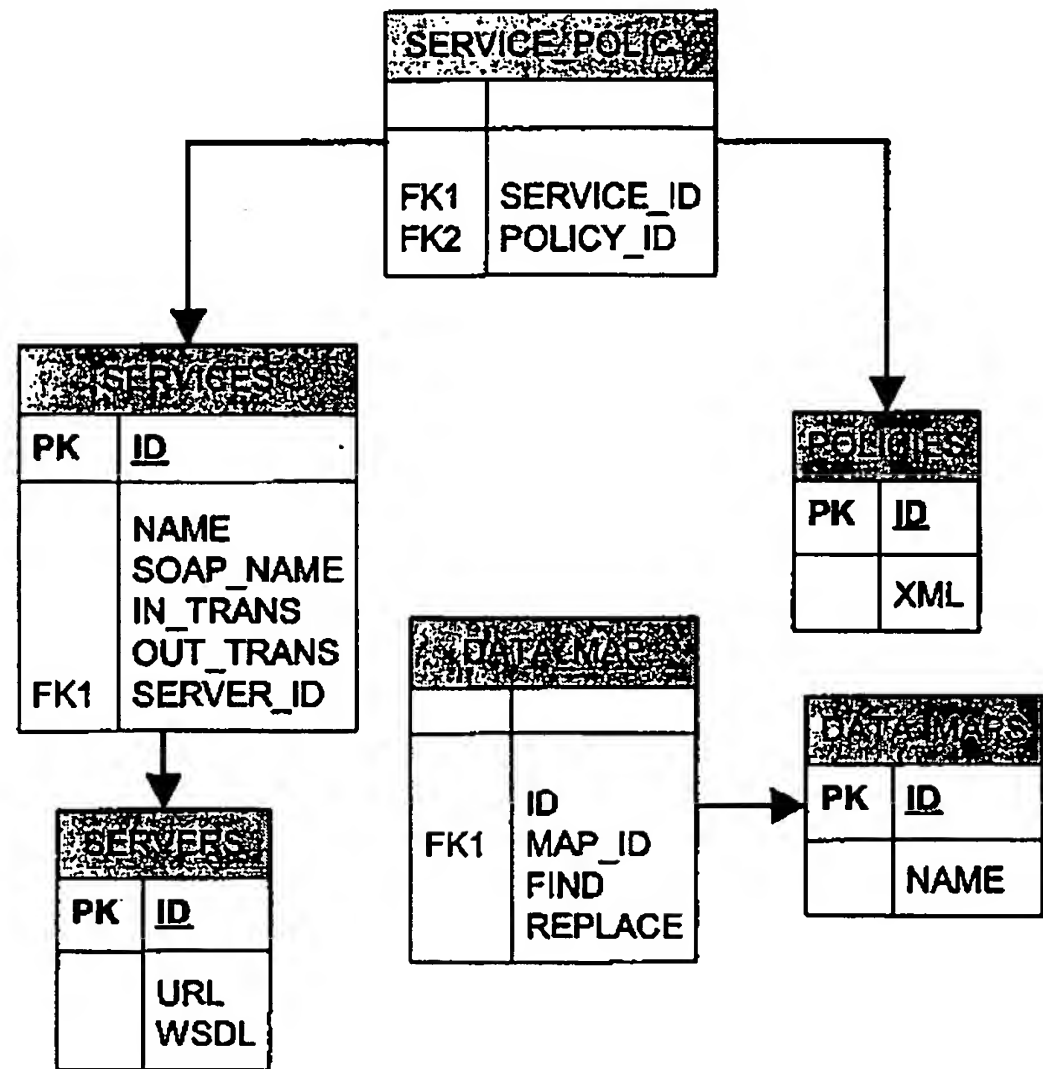
4.4 Policies



5. Deployment View

Stele SOAP Traffic Manager	Version: 1.0
Software Architecture Document	Date: 03.09.2001
<document identifier>	

6. Data View



The above diagram represents the tables stored in the SQL server. Description of the fields follows.

SERVICES	
NAME	Name of the service
SAOP_NAME	Name of the SOAP function to be called
IN_TRANS	XSLT of the transformation to be applied to incoming requests
OUT_TRANS	XSLT of the transformation to be applied to results
SERVER_ID	Foreign key in the table SERVERS

SERVERS	
URL	URL of the SOAP interface (HTTP transport assumed for demo, in final version this table will be enhanced)
WSDL	URL of the WSDL document describing the service

Stele SOAP Traffic Manager	Version: 1.0
Software Architecture Document	Date: 03.09.2001
<document identifier>	

POLICIES	
XML	Contains description of the policy in XML format. For a detailed description of the format follows.

6.1 Policy definition format

For the definition of policies XML-based format will be used. The following example describes the policy "If the service WithdrawCredit is called with parameter Amount greater than 10000 then send mail to Administrator and fail request.

```
<conditions>
  <cond name="ServiceEquals">
    <param value="WithdrawCredit"/>
  </cond>
  <cond name="Compare" op="gt">
    <param value="Amount"/>
    <param value="10000"/>
  </cond>
</conditions>
<actions>
  <action name="SendMail">
    <param value="Administrator"/> This is a role chacked in the LDAP
  </action>
  <action name="FailRequest">
    <param value="http://server.com/ammount_not_allowed.xml"/> Source for
returned XML result
  </action>
</actions>
```

For the prototype the supported conditions and actions and their syntax and parameters are described below.

6.1.1 Conditions

Service name equals a specific service

```
<cond name="ServiceEquals">
  <param value="name_to_match"/>
</cond>
```

Parameter [param] [op] [value]

eq	equals
lt	less than
gt	greater than
contains	contains

```
<cond name="Compare" op="op"> op = eq | lt | qt | contains
```


Stele SOAP Traffic Manager	Version: 1.0
Software Architecture Document	Date: 03.09.2001
<document identifier>	

```

        <param value="Param_to_compare"/>
        <param value="Value_to_compare"/>
    </cond>

```

6.1.2 Actions

Send mail to [person]

```

<action name="SendMail">
    <param value="person"/> This is a role chacked in the LDAP
</action>

```

Fail Request

```

<action name="FailRequest">
    <param value="http://server.com/ammount_not_allowed.xml"/> Source for
returned XML result
</action>

```

Store message in Log

```

<action name="LogMessage">
    Text to store in log
</action>

```